

Using open source software for antenna design: a short review

Gian Luigi Gragnani
DITEN – University of Genoa
Via Opera Pia 11A
16145 Genova, Italy
gianluigi.gragnani@unige.it

Claudio Montecucco
Atel Antennas s.r.l
Viale Caduti sul Lavoro 35
15061 Arquata Scrivia, Italy
claudio.montecucco@tin.it

Index Terms—Open source, electromagnetic simulation, antennas.

Abstract—In this paper a review of some open source programs for antenna design is carried out. Furthermore some results are presented and discussed also in comparison with the results obtained by means of commercial software.

I. INTRODUCTION

The present work has the main purpose of identifying possible open source programs that can be usefully employed for the design of antennas and possibly other passive electromagnetic devices such as, for example, filters, couplers, impedance adapters. The research also aims at finding software that allows the pre- and post-processing of data. Furthermore, it was decided to consider the possibility of easily integrating the design into the complete production and testing cycle.

II. OPEN SOURCE AND FREE SOFTWARE PROGRAMS

It is necessary to make a distinction, which is of considerable importance in the present context, between open source codes and free programs. The former, of which we will deal extensively, are also released in the form of source code and are freely modifiable and adaptable to the user needs (while some restrictions may exist for the redistribution of the modified program), while the latter are provided only in binary form and therefore do not allow any intervention, integration, porting, etc. by the user. The document is focused on open source programs, hence other programs, which could be useful but are available in binary form only, are not discussed here.

III. PROJECT FLOW

Without going into detail, in general in the flow of an antenna project, or, more generally, of any device, a sequence of three steps can be recognized. According to commonly used terminology, we distinguish:

- Pre-processing;
- Solution, using a calculation "engine" (solver);
- Post-processing.

A. Pre-processing phase

In the pre-processing phase we move from the idea of the real object to its approximate modeling, which is compatible with the calculation engine that will be used. The pre-processing phase can in turn be broken down into several steps, the most significant of which are:

- description of the geometrical and physical parameters of the project;
- approximation through a modeller, through the generation of a regular or irregular grid, which breaks down the device into many elementary blocks;
- generation of input data for the calculation engine.

The development of these steps may take place through the use of a graphical interface, or by writing command files in a special language, or, again, with a mix of the two possibilities. During the pre-processing phase, the documentation (drawings, layout, list of components, etc.) needed for production could also be generated.

B. Solution phase

In this phase the data generated during the pre-processing are supplied to the solver input, to obtain the numerical solution of the problem. The solver, in turn, will generate one or more result files, to be used in post-processing.

C. Post-processing phase

In the post-processing phase, the required data is extracted from the result produced by the solver. Also in this case the operations can be guided by a graphical interface or by command "script".

The data thus collected can be organized in files, tables, graphs of various kinds and will generally be used for different purposes. The most obvious is the verification of the result, but generally they will also be used to generate part of the documentation, both internal and public, to organize comparisons with other simulations or with similar devices, to generate input files for other types of simulations (for example radiation of a single antenna could be used in a calculation system for the study of arrays) and more.

If it has not already been produced during the pre-processing phase, at this stage the documentation necessary for the production of the device should also be generated.

IV. NUMERICAL METHODS

The most commonly used calculation methods for electromagnetic simulation can be divided into various classes, depending on the numerical approximation that is made starting from the Maxwell equations. Furthermore, there are methods that work directly in the time domain, while others work at single frequency.

Not all possible numerical methods are implemented in an open-source version. For antennas, in particular, the choice focuses essentially on two methods:

FDTD Method of Finite Differences in the Time Domain;
MoM Moment Method (implementation in the frequency domain).

A third method, very widespread also in the open source world for the solution of complicated equations, namely:

FEM Finite Element Method,
currently does not seem to be mature enough for an open application in electrodynamics.

V. COMMERCIAL PRODUCTS AND OPEN-SOURCE PRODUCTS

While it is to be dispelled the widespread opinion that open-source products are still worse than their commercial analogues, this is unfortunately true with regard to electromagnetic simulation. This is mainly due to the fact that the sector is relatively young and in any case rather niche, even for the open-source community, so the development of complete packages for electromagnetic calculation is slow. However, there are some packages that are already usable in a profitable way and are constantly developed.

A. Advantages and disadvantages of commercial products

In general, high-level commercial products contain all the tools necessary for the entire project flow. On the other hand, they present high costs, both at the time of purchase and for the maintenance of licenses. A list, even if not exhaustive, of salient points, for and against commercial packages, is reported in Figure 1.

B. Advantages and disadvantages of open-source products

At present, for electromagnetic simulation, the use of open-source software involves having to integrate non-homogeneous products, that usually do not offer the same level of functionality. To build a complete system, ranging from pre-processing to post-processing, it will certainly be necessary to integrate the various software with other parts, which need to be developed “at home”. On the other hand, access to the sources and the existence of a community of developers that is usually very available, sometimes allows problems to be solved much faster than for commercial products. A list, even if not exhaustive, of salient points, for and against open source packages, is reported in Figure 2.

VI. METHOD OF MOMENTS

The method of moments (MoM) is historically the one for which the first open source applications exist. The most well-known code that implements the method of moments is the NEC-2 [1]. Most of the MoM applications are developed in the frequency domain and the open source world is not an exception: all the programs described in Sect. VI-A are developed starting from the Maxwell equations in the frequency domain.

A. NEC-2 and derived programs

The reference program for many open source applications of the method is definitely the NEC-2. This program has been and still is used successfully for the analysis and design of many types of antennas. The NEC-2 is however limited to the modeling of metal wire/pipe structures, possibly connected through networks of lumped components or transmission lines, operating in a vacuum. The possibility of modeling antennas and devices containing dielectric parts is therefore excluded. When metal plates are to be modeled, they are approximated with a grid of wires. There are some limitations in the positioning of the intersections and the mutual positions of the wires, which have been highlighted over the years. The source code of the NEC-2, developed in Fortran, is still available, as are porting of the code in C [2] and C++ [3].

At present, there are no modelers or other programs available for generating input data, and the possibility of processing output data through open source programs is also rather limited. On the other hand, many prepackaged models of antennas of various kinds can be found in the Internet, which can serve as a starting point for a new project.

Among the most interesting software using the NEC-2 engine, we highlight:

1) *nec2c*: *nec2c* [2] is a porting in C of the original code; has been extensively tested and can easily replace the original. Compared to the Fortran code it allows the input and output to the command line and dynamically allocates the resources and therefore it is not necessary to recompile the program if the defined memory was insufficient.

2) *necpp*: *necpp* [3] is the C++ port of the original code; it can be interesting because the program has been restructured and is organized around a library that can be called by C, C++, Python and Ruby

3) *xnecview*: *xnecview* [4] is a visualization program for input and output: in input it allows to visualize the structure under examination, while the output provides the gain diagram in 2 and 3 dimensions, the input impedance, the VSWR and the maximum gain with respect to the frequency. It allows to export graphics as images, but not to create files and tables containing numeric data. The program code is rather cryptic.

4) *xnec2c*: using part of the *xnecview* code, the *nec2c* developer has created the program *xnec2c* [5], that allows to view and modify the project in a more or less interactive way. Unfortunately the program is only visual and is missing an output in numeric format. However, in the latest versions, it

Commercial Products

Pros:

- Complete and easy to use graphical user interface (GUI)
- Integrated processor
- Ability to import and export in different formats
- Usually different solvers available, to be chosen depending on the problem
- Solvers are in many cases able to exploit the maximum capabilities of the hardware
- Integrated in the package there is often also a programming language, for more complex elaborations
- Parametric optimizers, for finding the best solution with respect to the quantities of interest
- Post processing able to easily generate graphs and images and also, in certain cases, to generate written documentation
- Extended documentation
- Usually a certain number of calls to the help-desk are included in the price

Cons:

- High initial cost
- Generally the cost is for blocks of capacity of the tool, so each addition to the basic module must be paid separately
- Usually the licenses must be renewed annually
- Each machine must have its own license
- Often also using PCs with attached GPUs and/or in parallel mode, involves additional licensing fees
- If a machine fails, you must request, with valid reasons, to move the license to another machine
- It is not always easy to integrate the product with other software products that are external to the package
- If a feature not present in the software is needed, it is not easy to integrate it using external tools
- The file format needed for the program flow is often proprietary and undocumented, which makes it difficult to prepare input files or read output files with external tools

Figure 1. Pros and cons of commercial software tools

Open source tools

Pros:

- No license fees
- It is possible to install the software on different machines and possibly also in different versions
- Easy to maintain hardware backup
- Availability of the source code, so the program could in theory be modified or integrated with user-specific features
- Tools are usually programmable with standard programming languages and available on different operating systems
- Integration with other software is a typical feature of open source products
- Possibility to use various and sophisticated post processing tools
- Relatively easy to generate graphs, tables and documentation in text format
- The file format needed for the program flow is known
- The software developer community is usually available for dialogue and suggestions

Cons:

- Graphic user interface absent or very rudimentary
- Usually each phase of the project must be delegated to one or more different software, which must be integrated with each other by the user
- Powerful CAD tools and mesh generators for pre-processing exist, but they are not easy to use
- In the case of FDTD the generation of the mesh with the use of graphic tools is rather difficult
- While the ability to import and export in some standard formats is usually present, the interface to some commercial proprietary formats is absent also in high quality products
- Few reliable and easy to use solvers, and limited to the implementation of a single numerical method
- There are very sophisticated solvers, able to implement different numerical problems, but their complexity is such as to discourage their use
- Parametric optimization is possible, even with sophisticated methods, but only with external tools whose integration into the solution flow must be done by the user
- Difficult, or not convenient, to have a graphical and immediate display of the evolution of the processing
- Documentation often incomplete, obsolete or completely absent

Figure 2. Pros and cons of open-source tools.

is possible to export some data not only as images, but also as files that can be managed via gnuplot [6].

The NEC-2 code has evolved in versions 3 and 4. These codes have been classified for a long time and are not currently available for free, however it seems that the classification constraints have been removed almost completely and that now the NEC-4.2 can be obtained in source form (for a fee) [7].

B. MiniNEC3

Despite the similar name, the MiniNEC program is structurally different, both in code as well in input/output, compared to the NEC-2. The original code, written in an old version of BASIC, is still available today [8]. According to the users, the behavior of MiniNEC is better, compared to the NEC-2, for the management of closed space wires as well as in modeling junctions between wires of different diameters.

C. Puma-EM

Puma-EM [9] is a software developed for the study of the RCS (Radar Cross Section) of perfectly conductive objects. Therefore it does not have the necessary characteristics to model transmission antennas and to obtain the input parameters, however it has the interesting feature of accepting, as input geometric models, files generated with the Gmsh [10] program.

VII. FINITE DIFFERENCES TIME DOMAIN

Most recently developed open source software is based on the FDTD method. The method apparently offers two great advantages compared to the MoM: it does not require the inversion of large data matrices (and therefore it is relatively limited in the memory request) and, with a single simulation of the time domain, is able to evaluate the antenna over a wide range of frequencies. The method also has disadvantages, including the need to approximate the model on regular grids, however it is widely used, even in commercial products. Among the most interesting open source implementations are certainly OpenEMS [11], GprMax [12] and Meep [13].

A. OpenEMS

The openEMS code [11] is actually the solver of a well integrated suite of programs [14] for geometric modeling, resolution, output of data in various formats and integration with circuit simulation of electromagnetic devices and systems. Furthermore, there are some programs, notably pcb-rnd [15], that can export in a form suitable for openEMS.

B. gprMax

The gprMax program [12], as the name suggests, was born to simulate GPR (Ground Penetrating Radar) applications, but it is today a general purpose simulator. The development is very accurate [16] and much attention is paid to the theoretical correctness of the algorithms used. GprMax is one of the two programs, among those discussed in this document, that is currently able to exploit the computing power of a GPU (currently only Nvidia).

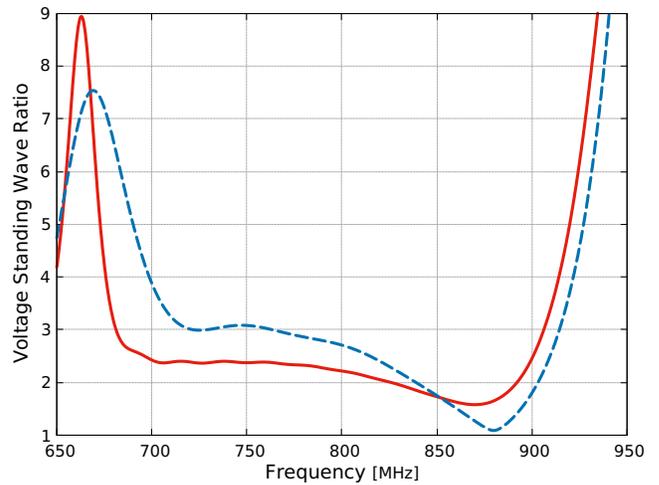


Figure 3. Comparison of the VSWRs, for a 3-element printed log-periodic antenna. Red continuous line: OpenEMS; Blu dashed line: commercial code.

C. Meep

Meep [13] is a program created in the field of advanced optics and is probably the most advanced of the open source FDTD programs. It is completely programmable and has some tools for post-processing. Nevertheless, because of its complexity, Meep is not easy to learn.

D. Other FDTD programs

There are some other software packages that are regularly maintained by the open source community and are worth citing: Vulture [17], GSvit [18], that can use GPU cards, and Angora [19].

VIII. FINITE ELEMENTS METHOD

As already mentioned, FEM codes for EM simulations are rather immature. However, there is a very interesting and promising project, named ONELAB, that is worth considering. At present, the project is based on Gmsh and GetDP [20].

IX. SIMULATIONS AND COMPARISONS

In order to check the capabilities of open source we designed some basic simulations, and compared the results among them and with those achieved by using a widespread commercial code.

Due to the limited space, we focused our attention to two tools only, that, in our opinion, are worth using in an open source platform: openEMS and nec2. In the first example, a 3-element printed log-periodic antenna was considered. This antenna had already been simulated by using a commercial tool, during a previous study [21]. In Fig. 3 the computed Voltage Standing Wave Ratios are shown. It can be seen that, despite some differences, the overall match between them is very good.

Another test aimed at comparing the behavior of the two open source tools examined. To this end, the BiQuad antenna that can be found on the tutorial examples of openEMS, was

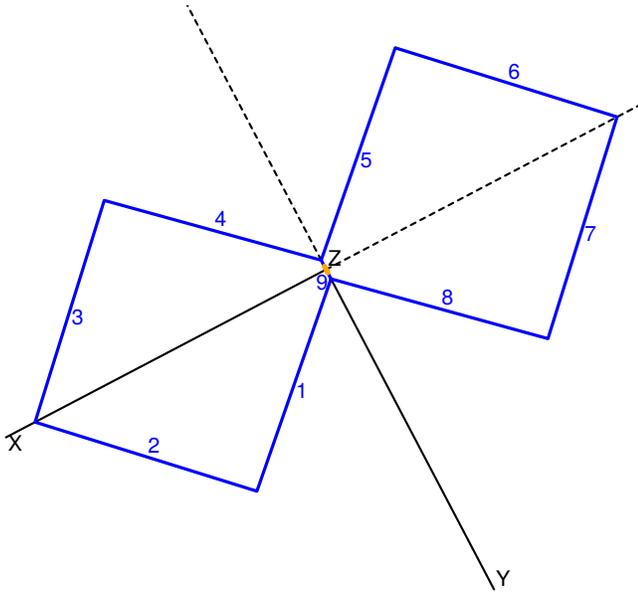


Figure 4. A pictorial view of the BiQuad antenna. The picture was extracted from the nec2c input file, by using xneccview.

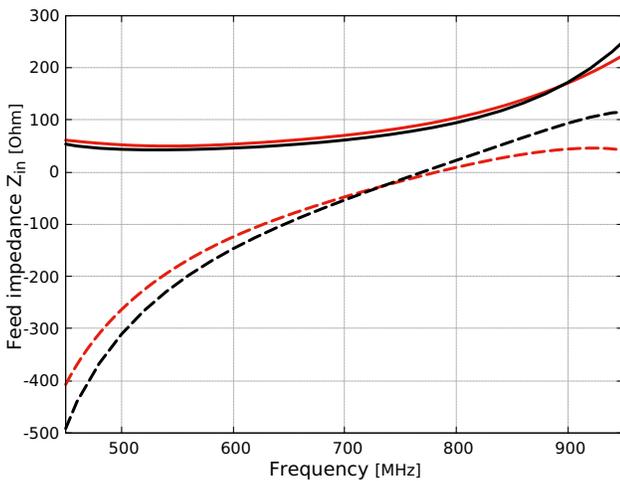


Figure 5. Comparison of the results obtained by openEMS and nec2c. Feed input impedance of the BiQuad antenna. Red line: openEMS; Black line: nec2c. Continuous lines: $\Re(Z_{in})$; Dashed lines: $\Im(Z_{in})$

simulated also by using nec2c. In Fig. 4, a schematic view of the antenna, extracted from the nec2c input file, by using xneccview, can be seen. In Fig. 5 the comparison of the input impedances given by the two simulators are given. Also in this case there is a very good agreement.

X. CONCLUSIONS

In this paper some consideration about open software for electromagnetic design, together with a short review of the possible solutions that are available, have been presented. While it is evident that commercial programs are still superior on managing the whole flow of the design and can offer useful utilities to produce data for production (component list, mechanical draws etc.) the work has focused that, in the

open source world, there are some very interesting packages for electromagnetic simulation, that can be satisfactorily used for the design of antennas or other electromagnetic devices. Furthermore, open source offers a wide variety of very high quality tools for pre- and post-processing of data. Hence, while the integration among the various open source packages is still at an infancy, open source software can nowadays be considered as a valid alternative to commercial tools.

REFERENCES

- [1] G. J. Burke and A. J. Poggio, "Numerical electromagnetic code (NEC)—Method of moments," Lawrence Livermore National Laboratory, Tech. Rep., Jan. 1981.
- [2] N. Kyriazis, "nec2c." [Online]. Available: <https://www.qsl.net/5b4az/pkg/nec2/nec2c/>
- [3] T. Molteno, "necpp." [Online]. Available: <https://github.com/tmolteno/necpp>
- [4] P.-T. de Boer, "xneccview." [Online]. Available: <https://wwwhome.ewi.utwente.nl/%7Eptdeboer/ham/xneccview/>
- [5] N. Kyriazis, "xnecc2c." [Online]. Available: <https://www.qsl.net/5b4az/pkg/nec2/xnecc2c/>
- [6] T. Williams, C. Kelley, R. Lang, D. Kotz, J. Campbell, G. Elber, and many others, "Gnuplot: an interactive plotting program." [Online]. Available: <http://gnuplot.sourceforge.net/>
- [7] "NEC (Numerical Electromagnetic Code)." [Online]. Available: <https://ipo.llnl.gov/technologies/nec>
- [8] [Online]. Available: <http://wb0dgd.com/mininec3.zip>
- [9] [Online]. Available: <http://puma-em.sourceforge.net/index.html>
- [10] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 9 2009. [Online]. Available: <https://doi.org/10.1002/nme.2579>
- [11] T. Liebig, A. Rennings, S. Held, and D. Erni, "openEMS – a free and open source equivalent-circuit (EC) FDTD simulation platform supporting cylindrical coordinates suitable for the analysis of traveling wave MRI applications," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 26, no. 6, pp. 680–696, 2013. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jnm.1875>
- [12] C. Warren, A. Giannopoulos, and I. Giannakis, "gprmax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," *Computer Physics Communications*, vol. 209, pp. 163 – 170, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S001046516302533>
- [13] A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. Joannopoulos, and S. G. Johnson, "Meep: A flexible free-software package for electromagnetic simulations by the ftdt method," *Computer Physics Communications*, vol. 181, no. 3, pp. 687 – 702, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S001046510900383X>
- [14] T. Liebig. openems - open electromagnetic field solver. General and Theoretical Electrical Engineering (ATE), University of Duisburg-Essen. [Online]. Available: <https://www.openEMS.de>
- [15] T. Palinkas. Pcb-rnd. [Online]. Available: <http://repo.hu/projects/pcb-rnd/index.html>
- [16] C. Warren, A. Giannopoulos, and I. Giannakis, "gprmax." [Online]. Available: <http://www.gprmax.com/>
- [17] I. D. Flintoft, "Vulture FDTD Code User Manual." [Online]. Available: <https://bitbucket.org/uoyaeg/vulture>
- [18] P. Klapetek. Gsvit. Department of nanometrology and technical length, Czech Metrology Institute. [Online]. Available: <http://gsvit.net/>
- [19] I. R. Capoglu and D. Zhang, "Angora: A finite-difference time-domain simulation package." [Online]. Available: <http://www.angorafdd.org/>
- [20] P. Dular and C. Geuzaine, "GetDP reference manual: the documentation for GetDP, a general environment for the treatment of discrete problems," <http://getdp.info>.
- [21] G. L. Gragnani, D. D. Caviglia, and C. Montecucco, "Design of printed loop-periodic antennas for long range communication within a wireless sensor network for sea water quality monitoring," *Automatic Control and Computer Sciences*, vol. 53, Aug 2019.